AFIN 2018

The Tenth International Conference on Advances in Future Internet

ISBN: 978-1-61208-662-0

September 16 - 20, 2018

Venice, Italy

AFIN 2018 Editors

Eugen Borcoci, University Politehnica of Bucharest, Romania

AFIN 2018

Forward

The Tenth International Conference on Advances in Future Internet (AFIN 2018), held between September 16, 2018 and September 20, 2018 in Venice, Italy, continued a series of events dealing with advances on future Internet mechanisms and services.

We are in the early stage of a revolution on what we call Internet now. Most of the design principles and deployments, as well as originally intended services, reached some technical limits and we can see a tremendous effort to correct this. Routing must be more intelligent, with quality of service consideration and 'on-demand' flavor, while the access control schemes should allow multiple technologies yet guarantying the privacy and integrity of the data. In a heavily distributed network resources, handling asset and resource for distributing computing (autonomic, cloud, on-demand) and addressing management in the next IPv6/IPv4 mixed networks require special effort for designers, equipment vendors, developers, and service providers.

The diversity of the Internet-based services requires a fair handling of transactions for financial applications, scalability for smart homes and ehealth/telemedicine, openness for web-based services, and protection of the private life. Different services have been developed and are going to grow based on future Internet mechanisms. Identifying the key issues and major challenges, as well as the potential solutions and the current results paves the way for future research

We take here the opportunity to warmly thank all the members of the AFIN 2018 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated their time and effort to contribute to AFIN 2018. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also gratefully thank the members of the AFIN 2018 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that AFIN 2018 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the field of future internet. We also hope that Venice, Italy provided a pleasant environment during the conference and everyone saved some time to enjoy the unique charm of the city.

AFIN 2018 Chairs

AFIN Steering Committee

Renwei (Richard) Li, Future Networks, Huawei, USA
Eugen Borcoci, University Politehnica of Bucharest, Romania
Alex Galis, University College London, UK
R.D. van der Mei (Rob), Centre for Mathematics and Computer Science (CWI), the Netherlands
Jun Peng, University of Texas - Rio Grande Valley, USA

Hiroyuki Sato, University of Tokyo, Japan
Sergio Ilarri, University of Zaragoza, Spain
Christos Bouras, University of Patras and Research Academic Computer Technology Institute, Greece
Adel Al-Jumaily, University of Technology, Sydney, Australia

AFIN Research/Industry Committee
Kiran Makhijani, Huawei Technologies, USA
Alexander Papaspyrou, adesso GmbH, Germany
Martin Zelm, INTEROP - Virtual Lab, Brussels, Belgium

AFIN 2018
Committee

AFIN Steering Committee
Renwei (Richard) Li, Future Networks, Huawei, USA
Eugen Borcoci, University Politehnica of Bucharest, Romania
Alex Galis, University College London, UK
R.D. van der Mei (Rob), Centre for Mathematics and Computer Science (CWI), the Netherlands
Jun Peng, University of Texas - Rio Grande Valley, USA
Hiroyuki Sato, University of Tokyo, Japan
Sergio Ilarri, University of Zaragoza, Spain
Christos Bouras, University of Patras and Research Academic Computer Technology Institute, Greece
Adel Al-Jumaily, University of Technology, Sydney, Australia

AFIN Research/Industry Committee
Kiran Makhijani, Huawei Technologies, USA
Alexander Papaspyrou, adesso GmbH, Germany
Martin Zelm, INTEROP - Virtual Lab, Brussels, Belgium

AFIN 2018 Technical Program Committee
Rocío Abascal-Mena, Universidad Autónoma Metropolitana - Cuajimalpa, Mexico
Cristina Alcaraz, University of Malaga, Spain
Muhammad Aleem, Capital University of Science and Technology (CUST), Pakistan
Adel Al-Jumaily, University of Technology, Sydney, Australia
Rafael Angarita, Isep Paris & Inria Paris, France
Rachida Aoudjit, Université Mouloud Mammeri de Tizi Ouzou, Algeria
Zubair Baig, Edith Cowan University, Australia
Marcin Bajer, ABB Corporate Research, Kraków, Poland
Paolo Bellavista, University of Bologna, Italy
Ana M. Bernardos, Universidad Politécnica de Madrid, Spain
Peter Bloodsworth, University of Oxford, UK
Eugen Borcoci, University Politehnica of Bucharest, Romania
Kechar Bouabdellah, University of Oran 1 Ahmed Ben Bella, Algeria
Christos Bouras, University of Patras and Research Academic Computer Technology Institute, Greece
Hongyu Pei Breivold, ABB Corporate Research, Sweden
Manuel José Cabral dos Santos Reis, University of Trás-os-Montes e Alto Douro, Portugal
Lianjie Cao, Purdue University, West Lafayette, USA
Kevin Chalmers, Edinburgh Napier University, UK
Fisnik Dalipi, Linnaeus University, Sweden
Laurenz Dallinger, Ludwig-Maximilians-Universität München, Germany
Maurizio D'Arienzo, Università della Campania Luigi Vanvitelli, Italy

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

A New Congestion Control Algorithm for Bandwidth Guaranteed Networks

Lin Han, Lijun Dong, Yingzhen Qu, Richard Li
Huawei USA – Futurewei Technologies, Inc.
Santa Clara, California, U.S.A
email: {lin.han, lijun.dong, yingzhen.qu, renwei.li}@huawei.com

***Abstract*— Future Internet can require bandwidth guaranteed services, thus the network resources need to be reserved before a Transmission Control Protocol (TCP) session starts transmitting data. The paper proposes a new TCP congestion control algorithm to assure the information rate for a flow. It is an extension to, yet different from the current TCP standards. The congestion window size changes at the sender side due to events, such as OAM (Operations, Administration and Management) congestion alarm, duplicate ACKs, and timeout.**

Keywords- QoS; TCP; IP; in-band signaling; bandwidth guaranteed networks; congestion control; congestion detection component.

I. INTRODUCTION

The current Internet, more generally a Transmission Control Protocol/Internet Protocol (TCP/IP) [1][2] network, was designed as a best-effort network, i.e., without any assurances as to Quality of Service (QoS), bandwidth, latency, processing time, etc. In other words, the IP service makes its "best effort" to deliver segments between two hosts, but it has no guarantees. On top of IP, TCP offers several additional services to applications. First and foremost, it provides reliable data transfer with flow control, sequence numbers, acknowledgments, and timers. TCP also provides congestion control [3], which prevents any one TCP connection from overwhelming the links and routers with a superabundant amount of traffic. TCP makes effort to provide each connection traversing an overloaded link with a fairly same proportion of the link bandwidth. The major components of congestion control in widely used TCP Reno include: slow start, congestion avoidance, and fast recovery. Figure 1. shows the congestion window at the sender side which changes over the time, as well as the state transfer due to the events.

With the unprecedented mobile applications emerging, such as Augmented Reality (AR) and Virtual Reality (VR), remote diagnosis and surgery, autonomous driving and road safety, guaranteeing the QoS in terms of bandwidth, latency, jitter etc. presents the unavoidable challenge to the current Internet's best-effort principle. In this paper, we use the term of "bandwidth guaranteed networks" to describe networks in which the bandwidth can be reserved for a particular flow. This can be achieved by the existing QoS mechanisms and frameworks: Integrated Services (IntServ) [4] with prior out of band signaling by RSVP [5], Differentiated Services Architecture (DiffServ) [6] with resource provisioning with the help of Service Layer Agreements (SLAs), Multiprotocol Label Switching (MPLS) [7] with Label Distribution Protocol (LDP) [8] and Resource Reservation Protocol-Traffic Engineering (RSVP-TE) and the in-band signaling protocol proposed in [9]. The common objective of all these solutions is to have network resources/bandwidth reserved before data is transmitted. In bandwidth guaranteed networks, the data transmission for a flow can be guaranteed at the committed information rate (CIR), but not above. When the data rate is between CIR and PIR (peak information rate), the shared resources are used. No traffic above PIR rate will be allowed to enter the network.

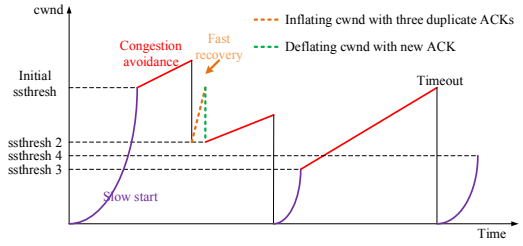

Figure 1. Congestion window in Reno

The paper proposes a new congestion control algorithm for the future Internet that builds upon TCP Reno, but considers the characteristics of bandwidth guaranteed networks as stated above. Section 2 explains the details of the new algorithm, and section 3 concludes this short paper.

II. NEW CONGESTION CONTROL ALGORITHM

The proposed congestion control algorithm has four components, which is introduced below. The congestion window size at the sender side is presented in Figure 2.

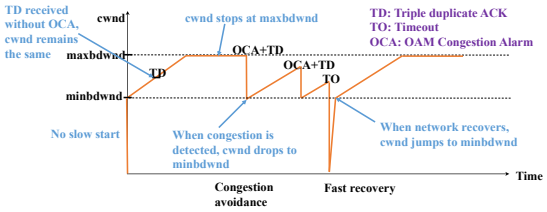

Figure 2. Congestion window in new congestion control algorithm for bandwidth guaranteed networks

A. Immediate Start

The proposed congestion control requires that OAM is used to constantly report on the network condition parameters, such as number of hops, Round Trip Time (RTT). This might be done through setting up a measuring TCP connection. The measuring TCP connection does not have user data, and it is only used to measure the key network parameters. As the network status is constantly changing, after a TCP session is

 ISBN: 978-1-61208-662-0

established, these parameters need to be updated. This requires a sender to periodically or consistently embed TCP data packet with OAM option. Consequently, in bandwidth guaranteed networks, the slow start component is not needed and removed from the proposed congestion control mechanism.

There are two important window sizes proposed for the new congestion control mechanism: *minbdwnd* and *maxbdwnd*, which are calculated as below:

$$minbdwnd = CIR * RTT/MSS \quad (1)$$

$$maxbdwnd = PIR * RTT/MSS \quad (2)$$

The RTT is the time taken to send a data packet to the destination and receive a response packet, and MSS is Maximum Segment Size. After a TCP session is established, the sender can immediately start transmitting data at an initial window size of *minbdwnd* as shown in Figure 2, if the receiver window (*rwnd*) is not a limiting factor.

Since the network status is constantly changing, RTT is updated using the following formula, with a is a number between 0 and 1:

$$RTT = a * old\ RTT + (1 - a) * new\ RTT \quad (3)$$

The initial RTT can be obtained by using a measuring TCP connection, or configured based on the historical data.

B. Congestion Avoidance

In bandwidth guaranteed networks, there is no slow-start, so congestion avoidance state is entered right after the initial start. During congestion avoidance, for every newly received ACK, *cwnd* is increased by one RTT/MSS until it reaches *maxbdwnd*. The value of *cwnd* stays constant at *maxbdwnd* afterwards, until packet loss is detected. This means a TCP sender is never allowed to send data at a rate larger than PIR.

C. Fast Retransmit and Fast Recovery

In the new congestion control algorithm, upon receiving duplicate ACKs the fast retransmit and fast recovery follow the following rules: (1) when a sender receives the first and second duplicate ACKs, the value of *cwnd* is not changed, and the sender continues to send traffic; (2) when a sender receives the third duplicated ACK, if the retransmission timer has not expired and a previous OAM congestion alarm has been received, it is likely a segment is lost due to congestion. The sender will perform a retransmission of the lost segment, and the value of *cwnd* is set to be *minbdwnd*; (3) when a sender receives the third duplicated ACK, but no previous OAM congestion alarm has been received, then it is considered that a segment is lost due to random failure instead of congestion. In this case, the value of *cwnd* is not changed.

D. Timeout Handling

If a retransmission timer in a TCP sender expires, in bandwidth guaranteed networks, this most likely indicates a physical failure no matter whether a duplicate ACK is received or not. In this case, the value of *cwnd* is set to be one, and the TCP sender will retransmit the lost segment. This retransmitted packet also serves the function of probing the network status. If there is really a network failure, no ACK will be received for this packet and the retransmission timer will expire again. Upon receiving an expected ACK after the retransmission(s), it indicates that the network has recovered from the physical failure, and the value of *cwnd* will be set to be *minbdwnd*.

III. SUMMARY AND OUTLOOK

A bandwidth guaranteed network is defined to be able to provide guaranteed bandwidth service with at least two bandwidth parameters: a Minimum Bandwidth or Committed information rate (CIR), and a Maximum Bandwidth or Peak information rate (PIR). The proposed congestion control algorithm to be used in bandwidth guaranteed networks comprises immediate start, congestion avoidance, fast retransmit and fast recovery, and timeout handling components. The detection of OAM signaling, duplicate ACKs, and timeout are used to infer the packet loss caused by random or permanent physical failure, or by congestion.

The proposed algorithm can coexist with current TCP congestion control mechanisms. Time sensitive TCP flows should achieve resource reservation before start sending data, and this guarantees bandwidth and latency especially when network is congested. Regular TCP sessions will share the remaining network resources.

In future works, we plan to implement the proposed congestion control in Huawei routers to prove the concept and verify that the guaranteed data rate of a flow can be achieved and not affected by other TCP flows. Moreover, we will extend the concepts and algorithms to realize guaranteed maximum latency for individual flow, which is extremely important and useful for latency sensitive applications.

REFERENCES

[1] S. Deering and R. Hinden, “RFC 8200: Internet Protocol, Version 6 (IPv6) Specification,” IETF, Jul. 2017.

[2] M. Allman, V.Paxson, and E. Blanton, “RFC 5681: TCP Congestion Control,” IETF, Sep. 2009.

[3] J. F. Kurose and K. W. Ross, “Computer Networking, A Topdown Approach”, Sixth Edition, ISBN: 9780132856201, 0132856204, published by Addison-Wesley Publishing Company.

[4] R. Braden, D. Clark, S. Shenker, “RFC 1663: Integrated Services in the Internet Architecutre: an Overview,” IETF, Jun. 1994.

[5] R. Braden, L. Zhang., S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP)-Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997.

[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “RFC 2475: An Architecture for Differentiated Services,” IETF, Dec. 1998.

[7] E. Rosen, A. Viswanathan, and R. Callon, “RFC 3031: Multiprotocol Label Switching Architecture,” IETF, 2001.

[8] L. Andersson, I. Minei, and B. Thomas, “RFC 5036: LDP Specification,” IETF, Oct. 2007.

[9] L. Han, Y. Qu, L. Dong, and R. Li, Flow-Level QoS Assurance via IPv6 In-Band Signalling, WOCC 2018.

 ISBN: 978-1-61208-662-0

Availability Assessment of IP Multimedia Subsystem in an NFV-based Environment

Mario Di Mauro[1], Giovanni Galatro[1], Maurizio Longo[1], Fabio Postiglione[1], Marco Tambasco[1,2]
[1]Dept. of Information and Electrical Engineering and Applied Mathematics (DIEM)
University of Salerno, Fisciano (SA), Italy
[2]Research Consortium on Telecommunications (CoRiTeL)
Via G. Paolo II, 84084, Fisciano (SA), Italy
email: {mdimauro, longo, fpostiglione}@unisa.it,
g.galatro1@studenti.unisa.it, marco.tambasco@coritel.it

***Abstract*—Network Function Virtualization (NFV) is considered one of the most influencing concepts in modern telecommunication frameworks, since it has the merit of transposing (and adapting) the virtualization paradigms from the computer world to the networking context. An instance of NFV is known as a Virtual Network Function (VNF), and represents a virtualized abstraction of a network element such as a router, a firewall, a load balancer, deployed in a virtualized environment. Actually, complex infrastructures, such as IP Multimedia Subsystem (IMS), a framework in charge of providing advanced multimedia services, can benefit of a virtualized deployment by implementing its constitutive elements as VNFs. The resulting architecture is a vIMS that, in this work, is characterized in terms of availability. More specifically, relying on a failure/repair model of a generic vIMS entity (modeled as a three-layer structure composed of hardware, hypervisor and software), we propose an availability assessment of the whole system by means of Stochastic Reward Networks framework.**

Keywords–Availability analysis; Stochastic Reward Networks; virtualized IP Multimedia Subsystem.

I. INTRODUCTION

Nowadays, telecom and network operators compete in deploying new services quickly and cheaply. Network Function Virtualization (NFV) [1] represents a valuable solution to face such issues, by implementing a *pay-per-use* model that allows to exploit a network service only as needed. According to this paradigm, a relocation or a hardware update of a traditional router, for example, can be replaced by manageable operations on a Virtual Network Function (VNF) exhibiting the same functionalities of the router itself. Generally speaking, a VNF can be represented by a three-layer structure composed of: a *hardware* layer representative of physical equipments (e.g., CPU, memory, etc.), a *hypervisor* layer serving as interface between hardware and software, and a *software* part representative of the particular VNF logic (e.g., routing, switching, etc.). In a similar manner, network elements of an IP Multimedia Subsystem (IMS) framework [2] can be recasted in terms of VNFs as pointed out in [3], [4], obtaining a virtualized IMS infrastructure denoted by vIMS. Starting from a vIMS exemplary architecture, in this work we advance a twofold contribution: first, we introduce a failure/repair model of a generic vIMS node compliant to the three-layer structure characterizing a VNF, and then, we perform an availability analysis of the resulting vIMS aimed at characterizing the optimal configuration that respects the "five nines" availability requirement, namely a maximum downtime tolerance of 5 minutes and 26 seconds per year. Such an assessment is obtained by application of Stochastic Reward Networks (SRN) framework when analyzing a single vIMS node, and, then, by considering the pipe of interconnected nodes by means of Reliability Block Diagram (RBD) representation. The paper is organized as follows: Section 2 contains a brief description of related research in the considered area. In Section 3, an overview about a vIMS deployment is offered. Section 4 introduces the availability model of a vIMS node, along with some details about the adopted methodologies (SRN and RBD). A numerical experiment useful to validate the considered model is proposed in Section 5, and, finally, concluding remarks end the work in Section 6.

II. RELATED WORK

In the field of telecommunication networks, availability issues are becoming crucial especially for those operators that have to obey some rigid Service Level Agreements. Besides, unlike the past, such issues have also to account for the massive presence of virtualized infrastructures characterizing modern telecommunication systems in cloud environments. Consequently, no wonder the technical and scientific literature is taking an interest about these aspects. Some valuable examples follow. Kim, Machida, and Trivedi in [5] propose one of the first availability models that consider the failure (and corresponding repair) events associated to the virtualization layer of a system, in addition to classical hardware and software failure actions. In particular, the authors largely exploit the Continuous-Time Markov Chain structures to model the behavior of some subsystems, such as CPU, memory, hypervisor, etc. A method useful to estimate some dependability attributes (availability among them) in virtualized environments has been proposed in [6], where the authors exploit the properties of Stochastic Petri Nets [7], a state-based model useful to account for redundancy strategies aimed at guaranteeing some availability requirements. The work presented in [8] is devoted at presenting a framework to evaluate the reliability of an NFV infrastructure where the focus is on some algorithms able to discover the minimum number of nodes that would cause the malfunctioning of the overall NFV deployment. In this case, the proposed model accounts for failure events but not repair actions. An approach based on the software rejuvenation applied to virtual environments and useful to cope with the occurrence of unplanned failures has been presented in [9] enriched with a detailed availability analysis, although hardware failures are not considered for simplicity. Another interesting approach is presented in [10], aimed at coping with novel container-based infrastructures by means of SRN

 ISBN: 978-1-61208-662-0

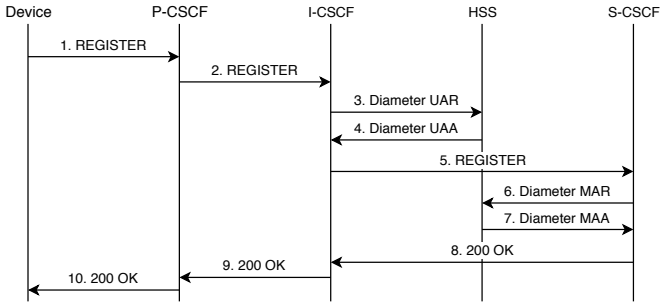

Figure 1. Registration procedure in IMS domain (simplified).

methodology. In line with this literature, in the present work we analyze a network infrastructure already considered in [11], [12], namely, a virtualized IMS framework, composed of hardware, hypervisor and software layers. However, differently from the previous work, here we adopt a double-layer availability model combining the expressive power of RBD, and the concise modeling offered by SRN.

III. IP Multimedia Subsystem overview

IMS enables a huge variety of architectures to provide multimedia services such as audio/video sessions, presence services, enriched communications. Furthermore, it has been elected as the reference architecture to support delivery of new voice services (e.g., Voice over LTE - VoLTE) across an all-IP network [13]. From an architectural perspective, IMS relies on a group of Call Session Control Function (CSCF) servers that communicate among them by means of Session Initiation Protocol (SIP) [14]. More specifically, the *Proxy* CSCF (P-CSCF) is the first contact point between a device and the IMS domain. The *Serving* CSCF (S-CSCF) represents the core of IMS and plays the role of a controller able to supervise critical aspects such as subscriber's service procedures or session status maintenance. The *Interrogating* CSCF (I-CSCF) acts as a gateway among multiple IMS domains determining whether or not the SIP messages forwarding is allowed from an operator to another. Finally, the Home Subscriber Server (HSS) is an evolved database which retains all the user data and is accessed by other CSCFs through Diameter protocol. For instance, when a user requests an access to the IMS domain, the S-CSCF queries to the HSS (via Diameter) to retrieve user profile in order to verify his/her grants. Typically, the message flow within an IMS domain follows a predefined path traversing a series of IMS nodes. It is the case of Registration procedure (depicted in Figure 1) where: a device requests to access the IMS domain by sending a REGISTER message to P-CSCF (1); such a message is passed to I-CSCF (2) that, in turn, queries the HSS the proper S-CSCF address that will manage the whole session. Such a query/response is identified by a couple of messages: User Authentication Request (UAR) (3), and User Authentication Answer (4). Once REGISTER message arrives to the S-CSCF (5), it retrieves user profile by the HSS through another couple of messages: Message Authentication Request (MAR) (6) and Message Authentication Response (7). If all goes well, the S-CSCF transmits a *200 OK* message to the device (8), (9), (10) and the registration procedure terminates.

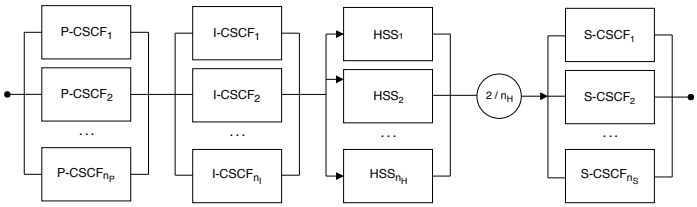

Figure 2. RBD representation of a vIMS domain, with HSS deployed in a 2-out-of-n_H redundancy configuration.

A. IMS within an NFV environment

IP Multimedia Subsystem can surely benefit from an NFV-based environment, by inheriting some advantages in terms of: *i)* flexibility: a vIMS element can be easily moved across geographical locations resulting in a cost-effective operation for a network provider; *ii)* manageability: a vIMS infrastructure can be effortlessly handled from a unified control center; *iii)* scalability: the hardware and software resources can be assigned to the vIMS framework in proportion to the real needs. Accordingly, an IMS node can be modeled as a three-layer structured VNF composed of:

- *Hardware (HW)*: typifies the physical components such as storage, CPU, memory, network etc.;
- *Virtual Machine Monitor (VMM)*: is the hypervisor, namely, the element which acts as an interface between hardware and software layers;
- *Software (SW)*: represents the application layer of each VNF which executes a specific functionality (X-CSCF, HSS, etc.).

In our scenario, two assumptions hold. First, hardware and hypervisor are reasonably supposed the same for all IMS elements recasted as VNFs. Secondly, the software layer admits different characterization for CSCF and HSS nodes.

IV. Availability Analysis

As previously stated, the availability analysis of a vIMS infrastructure performed in this work relies on a model which exploits two combined formalisms: RBD and SRN. The former offers a comfortable way to characterize the vIMS system by a macroscopic perspective, namely, in terms of high-level interconnections among nodes as depicted in Figure 2. In particular, the sketched representation embodies three aspects: *i)* a series model is used to characterize the chain of connections among the vIMS nodes; *ii)* a parallel configuration (per vIMS node) is representative of a redundancy strategy to cope with possible failures by assuming load balancing; *iii)* HSS element is supposed to be deployed in a *2-out-of-n_H* setting, meaning that 2 working HSS replicas are needed to consider HSS perfectly functioning. On the other hand, SRN methodology is exploited to model the interactions among the three layers (HW, SW, VMM) composing a generic vIMS node. The SRN framework [15] is a state-space model (derived from Markov Reward Models [16]) open to characterize a system in terms of its states distribution, by admitting a concise representation useful to mitigate the uncontrolled state space growth that typically occurs when dealing with classical probabilistic models. Basically, an SRN can be represented by a bi-partite directed graph with places (depicted by circles)

 ISBN: 978-1-61208-662-0

representative of conditions (e.g., the system is up or down), and transitions (depicted by rectangles) that account for actions (e.g., the system crashes or is restored). A place can contain a *token* (represented by a dot or a number) that indicates a particular holding condition, and that can be transferred to another place if a transition is *fired*, namely, if an action occurs. Transition times are supposed to be exponentially distributed and characterized by rates λ and μ associated to failure and repair actions, respectively. Evaluating an SRN means characterizing its *marking*, namely, its tokens distribution that changes across the time and provides information about system dynamics. From an analytical perspective, we are interested in evaluating the *reward function*, say $Z(t)$, a non-negative random process that can be associated to some relevant dependability metrics such as the availability. More specifically, the instantaneous availability obeys the following expression:

$$A(t) = Pr\{Z(t) = 1\} = E(Z(t)) = \sum_{i \in S} r_i \cdot p_i(t), \qquad (1)$$

where S represents the set of markings, split in a subset of up states (for which reward rate $r_i = 1$), and a subset of down states (for which $r_i = 0$), and where $p_i(t)$ denotes the probability of system being in state i. According to the three-layer model of a vIMS node, the corresponding SRN model of a vIMS node (either CSCF or HSS nodes) is as follows (see Figure 3):

- *Places* (circles): the set of places P_{upSW} [P_{dnSW}], P_{upVMM} [P_{dnVMM}], P_{upHW} [P_{dnHW}] accounts for the working [failure] conditions of software, hypervisor and hardware parts, respectively. The tokens within the "up" places are representative of initial working conditions.
- *Timed Transitions* (thick and unfilled rectangles): the set of transitions T_{fSW} [T_{rSW}], T_{fVMM} [T_{rVMM}], T_{fHW} [T_{rHW}] accounts for failure [repair] activities characterizing software, hypervisor and hardware parts, respectively.
- *Immediate Transitions* (thin and filled rectangles): the couple of transitions t_{SW} and t_{VMM} accounts for instantaneous actions occurring in an almost-zero transition time.

A. SRN model dynamics

Let study the SRN evolution of a generic vIMS node when failure and repair activities occur. Start from an initial working condition with 3 tokens in the three up places, consider the leftmost part of SRN in Figure 3. When a software failure occurs (e.g., the application part on top of CSCF or HSS node breaks) the token in P_{upSW} moves to P_{dnSW} as a consequence of fired transition T_{fSW}. The token will return in its original place (P_{upSW}) once a repair action occurs, namely, once T_{rSW} transition is fired. Instead, if a failure affects the hypervisor, the transition T_{fVMM} is fired, thus, the token leaves P_{upVMM} and arrives to P_{dnVMM}. In this case, an inhibitory arc (the segment between P_{upVMM} and t_{SW} with a little circle closer to the latter) becomes inactive and lets t_{SW} fire (no working software part is allowable when hypervisor fails). On the other hand, the inhibitory arc between P_{dnVMM} and T_{rSW} disables the latter by stopping the repair of the only software part when hypervisor is down (in other words, software and hypervisor repair is simultaneous through T_{rVMM}). Finally, upon a hardware layer failure, transition T_{fHW} is fired and the token, initially dwelling in P_{upHW}, is transferred to P_{dnHW}. In this case, the inhibitory arc between P_{upHW} and t_{VMM} entails the hypervisor failure once hardware fails, whereas, the arc connecting P_{dnHW} with T_{rVMM} avoids that a hypervisor repair activity be enabled until T_{rHW} is fired, namely, until the hardware is fixed. At this point we can define:

- $r_{i,k}$: reward rate pertaining to marking i for the k-th node replica;
- $p_{i,k}(t)$: probability of being in marking i at time t for the k-th node replica, computed by solving SRN in Figure 3 for each node.

Being all possible markings mutually exclusive, we can exploit (1) to derive the instantaneous availability $A^{(k)}(t)$ as

$$A^{(k)}(t) = \sum_{i \in I} r_{i,k} \cdot p_{i,k}(t), \qquad (2)$$

where I is the set of markings characterized by no immediate transitions enabled, and called *tangible markings*. Again, given marking i, the pertinent reward rate $r_{i,k}$ is defined as

$$r_{i,k} = \begin{cases} 1 & \text{if } (\#P_{upSW} = 1) \\ 0 & \text{otherwise,} \end{cases}$$

where $\#$ symbol denotes the number of tokens in a specific place. It is useful to notice that, such a condition does not account for "up" state of hardware and hypervisor, being basically contained in the SRN depicted in Figure 3 by means of inhibitory arcs By considering $\lim_{t\to\infty} A^{(k)}(t)$ we obtain the *steady-state availability* given by:

$$A^{(k)} = \lim_{t \to +\infty} A^{(k)}(t) = \sum_{i \in I} r_{i,k} \cdot p_{i,k}, \qquad (3)$$

where $p_{i,k}$ is the steady-state probability, namely $p_{i,k} = \lim_{t\to+\infty} p_{i,k}(t)$. By simple inspection of Figure 2, the vIMS infrastructure can be modeled by series/parallel interconnections among independent subsystems. Using (3), the steady-state availability of the whole vIMS system is given by:

$$A_{vIMS} = \left[1 - \prod_{k=1}^{n_P} \left(1 - A_P^{(k)}\right)\right] \cdot \left[1 - \prod_{k=1}^{n_S} \left(1 - A_S^{(k)}\right)\right] \left[1 - \prod_{k=1}^{n_I} \left(1 - A_I^{(k)}\right)\right] \cdot \sum_{k=2}^{n_H} \binom{n_H}{k} A_H^k \left(1 - A_H\right)^{n_H - k}, \qquad (4)$$

where:

- $A_P^{(k)}$, $A_S^{(k)}$, $A_I^{(k)}$ and $A_H^{(k)} = A_H$: steady-state availabilities of k-th replica of P-CSCF, S-CSCF, I-CSCF and HSS respectively;

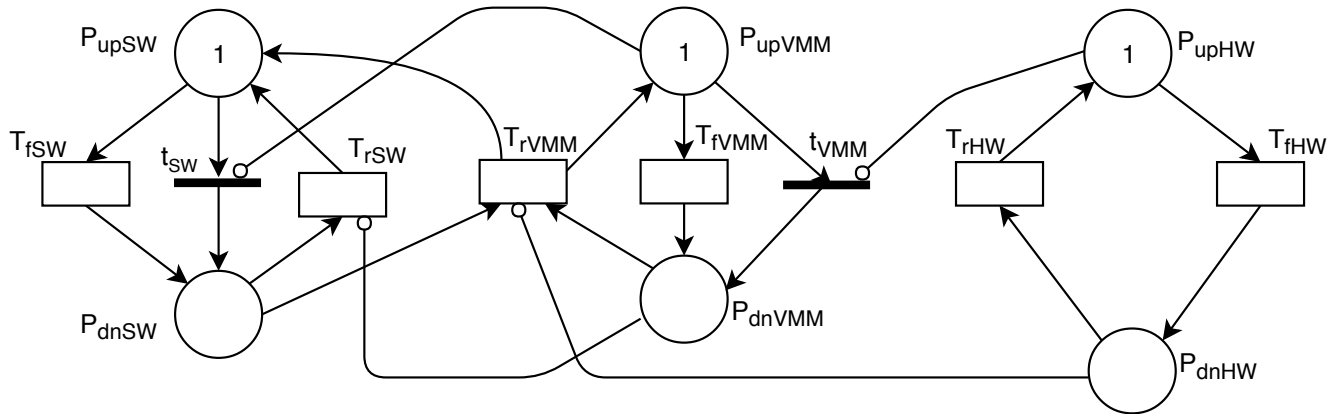

Figure 3. SRN model of a generic (either CSCF or HSS node) vIMS node according to the three-layer structure including: hardware (HW), virtual machine monitor (VMM) and software (SW).

TABLE I. INPUT PARAMETERS

Parameter	Description	Value
$1/\lambda_{HW}$	mean time for hardware failure	60000 hours
$1/\lambda_{VMM}$	mean time for hypervisor failure	5000 hours
$1/\lambda_{CSCF}$	mean time for CSCF node failure	3000 hours
$1/\lambda_{HSS}$	mean time for HSS node failure	2000 hours
$1/\mu_{HW}$	mean time for hardware repair	8 hours
$1/\mu_{VMM}$	mean time for hypervisor repair	2 hours
$1/\mu_{CSCF}$	mean time for CSCF software repair	1 hour
$1/\mu_{HSS}$	mean time for HSS software repair	1 hour

TABLE II. AVAILABILITY RESULTS OF VIMS BY CONSIDERING 5 EXEMPLARY SETTINGS $(S_1, \ldots, S_5)$.

Setting	Redundancy Level	A_{vIMS}
S_1	$CSCF = [2, 2, 2]$, $HSS = 2$	0.997867
S_2	$CSCF = [2, 2, 3]$, $HSS = 2$	0.997868
S_3	$CSCF = [2, 2, 2]$, $HSS = 3$	0.999994
S_4	$CSCF = [2, 2, 3]$, $HSS = 3$	0.999995
S_5	$CSCF = [2, 3, 3]$, $HSS = 4$	0.999999

- n_P, n_S, n_I and n_H: number of redundant subsystems of each functionality (P-CSCF, S-CSCF, I-CSCF and HSS respectively).

The steady-state availability in (4) appears as a product of the first three factors associated to the series of nodes P-CSCF, S-CSCF and I-CSCF replicated in a parallel configuration. The last term addresses the *2-out-of-n_H* scheme characterizing HSS node.

V. Numerical Example

By starting from the previously modeled vIMS framework, in this section we propose a numerical experiment with the support of an effective tool named SHARPE [17]. In particular, we perform an availability analysis aimed at identifying the optimal configuration respecting the "five nines" condition, by exploiting values reported in Table I (in line with [11]). We make two assumptions: first, we consider that software instances on top of CSCF nodes are characterized by the same failure and repair rates, with the exception of HSS database which is intrinsically prone to more faults. Second, we assume that hypervisor and hardware layers are the same for all nodes. The steady-state availability analysis is performed by considering different system configurations. We report here five exemplary settings $S_1, \ldots, S_5$ (among the tested ones) to show how the number of parallel nodes influences system availability:

- S_1: 2 replicas for each vIMS node (CSCFs and HSS);
- S_2: 2 replicas for a couple of CSCFs, 3 replicas for the remaining CSCF and 2 replicas for HSS;
- S_3: 2 replicas for each CSCF and 3 replicas for HSS;
- S_4: 2 replicas for a couple of CSCFs, 3 replicas for the remaining CSCF and 3 replicas for HSS;
- S_5: 2 replicas for a single CSCF, 3 replicas for a couple of CSCFs and 4 replicas for HSS.

The results are reported in Table II. Notice that S_1 and S_2 settings are far below the "five nines" availability requirement, due to the lack of any redundant node for HSS. On the other hand, S_3 and S_4 settings satisfy both the desired condition with 9 and 10 node replicas, respectively, thus, S_3 is preferable being more cost-effective. Finally, setting S_5, with 12 node replicas and two redundant nodes for HSS, allows to achieve a "six nines" availability condition which is required in some strongly critical infrastructures.

VI. Conclusions

Nowadays, network infrastructures can derive copious advantages from Network Function Virtualization (NFV) paradigm in terms of flexibility, scalability, cost saving and maintenance. A paramount example is represented by IP Multimedia Subsystem (IMS), the framework acting as core network for modern telecommunication infrastructures such as Voice over LTE (VoLTE) or Voice over Wi-Fi (VoWi-Fi). Such a framework is prone to adhere to the NFV standard by virtualizing its main nodes, namely, the CSCFs and the HSS. Accordingly, in this work we propose an availability analysis of a virtualized IMS infrastructure (that we call vIMS) performed through two formalisms: the Reliability Block Diagram (RBD) useful to characterize the high-level interconnections among vIMS nodes, and the Stochastic Reward Nets (SRN) helpful to model in a probabilistic way the failure/repair events occurring at any of the three layers (software, hypervisor,

 ISBN: 978-1-61208-662-0

hardware) of a vIMS node. Such an availability analysis can be easily afforded by exploiting well-assessed software tools (SHARPE) and results advantageous to identify the optimal vIMS configuration matching the "five nines" availability requirements.

Future works will take into account: more sophisticated performance models (with a view to the so-called performability analysis), more complex interconnections among the three-layer structure of a vIMS node, where a co-location of some nodes could be considered as is the case of more realistic scenarios, and fault injection methods aimed at characterizing more realistically the recovery time.

REFERENCES

[1] ETSI, "Network Functions Virtualisation: An introduction, benefits, enablers, challenges and call for action," Tech. Rep., 2012.

[2] G. Camarillo and M. Garcia-Martin, The 3G IP Multimedia Subsystem, 3rd ed. John Wiley and Sons, 2008, ISBN: 9780470516621.

[3] Ericsson Review, "Virtualizing network services - the telecom cloud," 2014 [Online], available: https://www.ericsson.com/en/ericsson-technology-review/archive/2014/virtualizing-network-services---the-telecom-cloud [accessed:2018-07-10].

[4] "Project clearwater," available: http://www.projectclearwater.org/ [accessed:2018-07-10].

[5] D. S. Kim, F. Machida, and K. S. Trivedi, "Availability modeling and analysis of a virtualized system," in Proc. IEEE PRDC 2009, 2009, pp. 365–371.

[6] S. Fernandes, E. Tavares, M. Santos, V. Lira, and P. Maciel, "Dependability assessment of virtualized networks," in Proc. IEEE ICC 2012, 2012, pp. 2711–2716.

[7] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, 1st ed. John Wiley & Sons, Inc., 1994, ISBN:0471930598.

[8] J. Liu, Z. Jiang, N. Kato, O. Akashi, and A. Takahara, "Reliability evaluation for NFV deployment of future mobile broadband networks," IEEE Wireless Communications, vol. 23, no. 3, 2016, pp. 90–96, ISSN: 15361284.

[9] T. Thein and J. Sou Park, "Availability analysis of application servers using software rejuvenation and virtualization," Journal of Computer Science and Technology, vol. 24, no. 2, 2009, pp. 339–346, ISSN: 18604749.

[10] S. Sebastio, R. Ghosh, and T. Mukherjee, "An availability analysis approach for deployment configurations of containers," IEEE Transactions on Services Computing, 2018, pp. 1–1, ISSN: 19391374.

[11] M. Di Mauro, G. Galatro, M. Longo, F. Postiglione, and M. Tambasco, "Availability evaluation of a virtualized IP Multimedia Subsystem for 5G network architectures," in Safety and Reliability - Theory and Applications, M. Cepin and R. Bris, Eds. Taylor & Francis Group, 2017, pp. 2203–2210, ISBN: 9781351809726.

[12] M. Di Mauro, M. Longo, F. Postiglione, and M. Tambasco, "Availability modeling and evaluation of a network service deployed via NFV," in Digital Communication. Towards a Smart and Secure Future Internet, A. Piva, I. Tinnirello, and S. Morosi, Eds. Springer International Publishing, 2017, pp. 31–44, ISBN: 9783319676395.

[13] Nokia Networks, "Evolve to richer voice with Voice over LTE (VoLTE)," 2014 [Online], available: https://onestore.nokia.com/asset/200306/Nokia_VoLTE_White_Paper_EN.pdf [accessed:2018-07-10].

[14] J. D. Rosenberg et al., "Session Initiation Protocol (SIP)," 2002, IETF RFC 3261.

[15] J. K. Muppala, G. Ciardo, and K. S. Trivedi, "Stochastic Reward Nets for reliability prediction," in Communications in Reliability, Maintainability and Serviceability, 1994, pp. 9–20.

[16] A. Reibman, R. Smith, and K. S. Trivedi, "Markov and Markov reward model transient analysis: An overview of numerical approaches," European Journal of Operational Research, vol. 40, no. 2, 1989, pp. 257–267, ISSN: 03772217.

[17] K. S. Trivedi and R. Sahner, "SHARPE at the age of twenty two," SIGMETRICS Perform. Eval. Rev., vol. 36, no. 4, 2009, pp. 52–57, ISSN: 01635999.

 ISBN: 978-1-61208-662-0

Mobile Edge Computing versus Fog Computing in Internet of Vehicles

Eugen Borcoci, Marius Vochin, Serban Obreja
University POLITEHNICA of Bucharest - UPB
Bucharest, Romania
Emails: eugen.borcoci@elcom.pub.ro, marius.vochin@upb.ro, serban@radio.pub.ro

***Abstract* —Vehicular networks and the recent Internet of Vehicles (IoV) are continuously developing, aiming to solve the current and novel challenging needs in the domain of transportation systems. Edge computing offers a natural support for Internet of Vehicles, supporting fast response, context awareness, and minimization of the data transfer to the centralized data centers - all these being allowed by the edge computing availability close to mobile vehicles. Multi-access (Mobile) Edge Computing, fog computing, cloudlets, etc., are such candidates to support IoV; their architectures and technologies have overlapping characteristics but also differences in approach. A full convergence between them has not yet been achieved. Also, it is still not completely clarified which solution could be the best trade-off to be adopted in the Internet of vehicles context and for which use cases. This paper is not a complete survey, but attempts a preliminary evaluation of some of the currently proposed Mobile Edge Computing and fog computing solutions for vehicular networks.**

Keywords — Internet of Vehicles; Vehicular Networks; Fog computing; Edge computing; Software Defined Networking; Network Function Virtualization.

I. INTRODUCTION

Vehicular communications, networks and associated services constitute a significant area of research, development and implementation in the framework of the *Intelligent Transport System* (ITS) [1] and, more recently, *Internet of Vehicles.* Supporting networking technologies have been developed, e.g., *Dedicated Short-Range Communications* (DSRC) and also higher functional layers such as the *Wireless Access in Vehicular Environments* (WAVE) [2]. The IEEE 802.11a/p and respectively IEEE 1609 represent a mature set of standards for DSRC/WAVE networks. For wide area, 4G, *Long Term Evolution* (Advanced) (LTE-A) are used and, in the future, 5G is a strong candidate.

Vehicular Ad Hoc Networks (VANET) [3] (a larger class will be denoted as VNET) generally VNETs have been defined to support basic communications types: vehicle to vehicle (V2V), - to road (V2R), or - to Infrastructure (V2I) in uni- or bi-directional mode (note that, some authors include V2R into V2I type). The basic VANET functional components are the *On-Board-Unit* (OBU), inside the vehicles and *Road-Side–Unit* (RSU) placed on the roads. The RSUs communicate with vehicles through wireless access and among them via external networks. The main use cases of VANET have been oriented to safety and traffic management.

Novel Internet of Vehicles architectures have been recently proposed to extend VANETs and aiming a global span of a vehicle network [4-7]. IoV can be also considered as a special case of *Internet of Things* (IoT) [8], where the "things" are either vehicles or their subsystems. The IoV connects the vehicles and RSUs through different *Wireless/Radio Access Technologies* (WAT/RAT), while traditional Internet and other heterogeneous networks cover the wide area. IoV aims to serve a large range of applications and use cases, including those coming from ITS, V(A)NETs and other novel ones:

- Safety and management oriented

Safety: emergency call, warnings (wrong-way, lane change, overtaking), automatic breaking, automatic speed control; *Traffic and navigation management*: real time traffic information, parked car and parking space locating, parking space offers and booking, speeding evidence, navigation area extension, multi-modal transportation, traffic signaling, localizing events, logging, etc.; *Remote telematics*: car surveillance, fuel usage optimization, remote locking/unlocking, stolen vehicle recovery, driving behavior analysis, diagnostic actions, etc.

- Business oriented

Infotainment: Wi-Fi in vehicle, content downloading, online radio and streaming, SMS, advertisements, calendar and address book, Facebook/WhatsApp, location sharing/tracking family/friends, connected drive; *Insurance*: group/family/usage/season/region-based; *Car Sharing*: booking for family/group car/group-parking, car pooling and sharing; *Other services*: cloud/fog/edge various services, mobile toll payment, driving behavior analysis, etc.

To develop the above applications, IoV can take benefit from centralized *Cloud Computing* (CC) combined with *Edge Computing (EC)* [9][10].

EC moves the cloud computing capabilities (computation, storage) at the network edge, thus offering for IoV more appropriate features than CC: faster time response, more flexibility in functional distribution, context awareness, resource usage optimization and reduction in the amount of data exchanged between a cloud data center and a vehicle.

We adopt here a general view, that EC can represent any set of computing and network resources distributed along the path between data sources and cloud data centers.

 ISBN: 978-1-61208-662-0

However, there is not yet a unique vision on "edge" semantics, except the common attribute of proximity to the data sources. Note that some studies see the EC as restricted to edge devices only (some edge nodes are defined, which can be composed of smart sensors, smart phones, and smart vehicles, even a special edge servers). Currently, significant overlap exists between particular EC architectures and convergence is predicted in the near future. However, several sets of EC specifications have been elaborated by independent organizations. The most relevant ones, are *Multi-access (Mobile) Edge Computing* (MEC) [11][12] and *Fog computing* (FC) [13]-[15]. Their deployment is strongly supported by industry and operators.

Virtualization technologies constitute an important "tool" in developing edge computing. In the architectural management and control planes, *Software-defined networking* (SDN) [16] and *Network Function Virtualization* (NFV) [17] are seen as a strong EC and IoV support, given their features like flexibility, programmability, abstraction via virtualization, dynamicity, etc. SDN decouples the data/user plane from control plane and logically centralizes the control. NFV moves into software many network functions that traditionally have been implemented by dedicated hardware and software. SDN and NFV can be applied independently, but their cooperation (they could be seen as "orthogonal") is considered as a powerful approach.

MEC, FC, cloudlets, etc., are attractive for V(A)NET and IoV. They have overlapping characteristics but also expose differences in approach. The best trade-off to be adopted in a specific IoV context is still not clear. This paper attempts a preliminary comparison of some MEC and FC solutions for IoV.

Note that, given the limited dimension of this paper, it is not claimed to be a complete survey on the topics. The analysis is high level only; the aim is not to detail certain functions or services, but to evaluate several variants of solutions and some guidelines for selection of an approach appropriate for IoV specific use cases. While they are important in vehicular environment, some topics are not discussed in this study, e.g., security and privacy issues.

The paper is organized as follows. Section II is a very short overview of MEC/Fog related work. Section III is a summary overview of IoV layered architectures. Section IV selects samples of solutions to realize VANETs/IoV in MEC or FC approaches, including some SDN/NFV based. Section V tries to identify some pros and cons for MEC and FC variants, while emphasizing the points of convergence. Section VI contains conclusions and possible future work.

II. Related Work: Edge Computing Architectures

This section presents a very short summary of the MEC and FC architectures; both of them can be considered as strong candidates to support IoV. More comprehensive descriptions can be found in references [10]-[15].

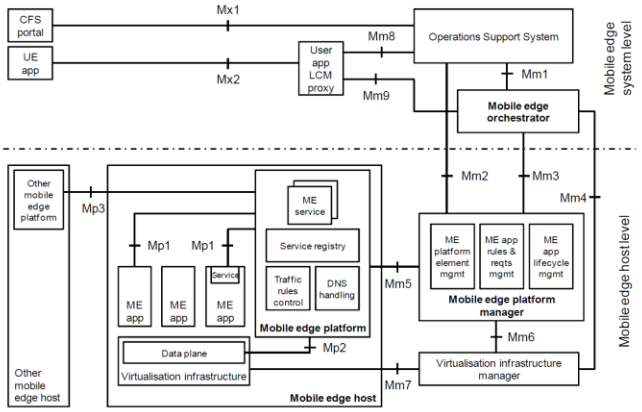

Figure 1. MEC reference architecture (ETSI)[11][12]

A. Multi-access(Mobile) Edge Computing

MEC architecture promoted mainly by ETSI [11][12], offers low latency/response time, high bandwidth, location and context awareness, reduction in amount of data transferred from/to a terminal device to a centralized cloud data center, etc.

ETSI has established in 2014 the MEC Industry Specification Group which provided first specifications. In 2017, the MEC name (and scope) has been extended to *Multi-access Edge Computing* [12] - to include non-cellular and fixed access cases. MEC supports multi-services and multi-tenancy; third-parties may also make use of the MEC storage and processing capabilities.

The MEC resources are placed at the network edge (e.g., in *Radio Access Network* – RAN, i.e., Base Stations, or in aggregation points, etc.). The key element is MEC *mobile edge host* (MEH) playing the role of an application server. It is integrated in RAN and provides computing resources, storage capacity, connectivity, and access to user traffic, radio and network information.

The MEC reference architecture is presented in Figure 1 (details, in [11][12]). The mobile edge host level is the main MEC sub-system, composed of: the mobile edge host (MEH) and its management. The MEH includes a virtualization infrastructure (based on *Network Function Virtualisation Infrastructure* – NFVI - coming from ETSI NFV framework) and the *mobile edge platform* (MEP), supporting the execution of mobile edge applications.

The MEC server can be installed in various places at the network edge: at the 4G/LTE macro base station (eNB); at the multi-technology (3G/LTE) cell aggregation site; at the Radio Network Controller (RNC) site, for 3G. MEC is seen as an efficient technology to support V(A)NET/IoV [4]-[7]. Vehicles connected to the distributed edges may send/receive information to/from other vehicles or through the network, almost in real-time.

B. Fog Computing

Fog computing (FC) [13]-[15] is another recent EC technology complementary to CC (FC will not replace the CC, but cooperation cloud/fog is envisaged). The FC distributed platform brings computation and storage close to

its data sources, to reduce the latency and cost of delivering data to a remote cloud. FC has been proposed originally to support the IoT, introduced by Cisco [13].

An important FC-related document is taken by the *OpenFog Consortium* (2015) [15]. That is why this section dedicatees more space to it. OpenFog consortium defines FC as a system-level horizontal architecture that distributes resources and services of computing, storage, control and networking *anywhere along the continuum* from a cloud data center down to things. On the other side, *MEC*, originally targets only the very edge part of the network (e.g., RAN). That is why, some authors consider MEC as a special case of FC.

FC can support multiple industry verticals and application domains delivering intelligence and services to users and business. FC capability is spanning across multiple protocol layers and is not dependent on specific access systems. FC focuses the processing efforts outside the cloud data center i.e., in the fog area. Data are gathered, processed, and stored within the network, by way of an IoT gateway (GW) or an FC node (FN). Information is transmitted to this GW from various sources and it is processed in FN; then pertinent data (plus additional command - if necessary), are transmitted back, towards the devices. A FN can process data received from multiple end points and send information exactly where it is needed.

OpenFog Consortium has defined a flexible deployment hierarchical model for FC, IoT oriented, as presented in Figure 2 [15]. Several use cases can be accommodated in this model.

The case 1 shows a FC-based only system (the CC cannot be used for some reasons like response time, special requirements, special transportation systems environments, unavailability, etc.). In the case 2, the operation-centric information processing is done by FNs located close to the infrastructure/process being managed while cloud processing is performed only for event-to-action time window ranging from hours to months. The case 3 shows the local fog infrastructure used for time-sensitive computation, while the cloud is used for the balance of operational and business-related information processing (commercial device monitoring, mobile network acceleration, content delivery networks – CDNs).

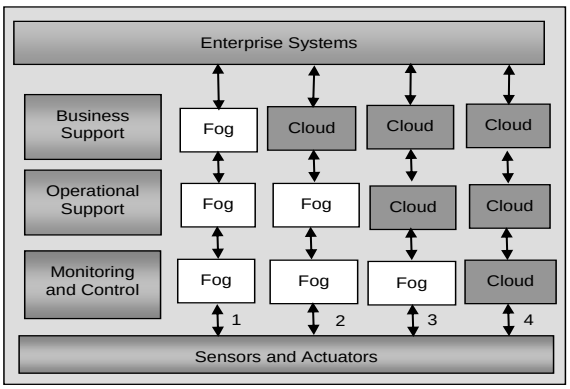

Figure 2. IoT System deployment models - variants [adapted from [14]]

The case 4 supports use cases like agriculture, connected cars, and remote weather stations. The cloud is used for the entire stack due to the constrained environments in which the deployment of fog infrastructure may not be feasible or economical. However, FNs at the device layer may get some of the monitoring and control functions for safety related control. The enterprise systems integrate with cloud for business operations.

Recently the work [18] extended the FC scope, by defining *Fog of Everything (FoE)* to serve *Internet of Everything (IoE).* The FNs are usually virtualized networked data centers, which run on top of (typically, wireless) *Access Points* (APs), at the edge of the access network, resulting in a three-tier IoE-Fog-Cloud hierarchy. In this context, a "thing" (fixed, nomadic or mobile) is a resource-limited user device that needs resource augmentation in order to execute its workload. The work [18] proposes a hierarchical general architecture for a FoE virtualized platform, integrating the building blocks:

- *IoE layer*, where a number of (possibly, heterogeneous) things operate over multiple spatial clusters;
- *wireless access* network (fixed/mobile), to support *Fog-to-Thing* (F2T) and *Thing-to-Fog* (T2F) communication through TCP/IP connections running atop, e.g., *IEEE802.11/15* single-hop links;
- a set of *inter-connected FNs*, that act as virtualized cluster headers;
- *inter-Fog backbone* (wireline/wireless) providing inter-Fog connectivity and making feasible inter-Fog resource pooling;
- *virtualization layer*, allowing things to augment their limited resources by exploiting the computing capability of a corresponding virtual clone. This last one runs atop a physical server of the FN that currently serves the cloned thing;
- the resulting *overlay inter-clone virtual network*, that allows P2P inter-clone communication by relying on TCP/IP E2E connections.

The corresponding protocol stack [17] comprises four layers:

IoE layer provides services like: *(a)* T2F access through a reservation-based collision free access protocol for the things served by a same FN; *(b)* F2T broadcast services.

Fog layer performs: *(a)* energy-efficient management of the networking and computing physical resources equipping each FN, and *(b)* energy-efficient management of the inter-Fog traffic conveyed by the wireless backbone.

Overlay layer supports the overlay inter-clone P2P network by: (a) inter-Fog clone migration; it can be supported by the implementation of the so-called *Follow-Me-Cloud* framework (e.g., Taleb et al., [18]), to solve "live" inter-Fog clone migration, in response to the thing mobility; (b) dynamic management of the required migration bandwidth, to minimize the energy consumed by clone migrations.

 ISBN: 978-1-61208-662-0

Cloud layer orchestrates the overall Cloud-Fog-IoE platform based on the specific features and requirements of the running applications. The solutions must be tailored on the expected attributes of the supported applications.

III. RELEVANT IOV LAYERED ARCHITECTURES

This sub-section shortly presents some recent IoV architectures, given that edge computing functionalities should be integrated in such IoV architectures. Some relevant ones have been selected.

Among the first proposals, there is Bonomi et al. [8] four-layered architecture, for connected vehicles and transportation. A layer includes groups of functions, which could be mapped on one or more classical TCP/IP layers. Also, the four layers correspond to different geo-locations of the subsystems (vehicles, networking infrastructure, cloud data centers, etc.). The bottom layer (L1-end points) represents the vehicles, and their communication protocols (basically for V2V communication, using the IEEE 802.11a/p). The L2 (infrastructure), represents communication technologies to interconnect the IoV actors (via WiFi, 802.11p, 3G/4G, etc.). The L3 (operation) performs management; it verifies and ensures compliance with all applicable policies, to regulate the information management and flow. The L4 (cloud- public, private or enterprise) is based on a defined profile coupled with the possibility of receiving services (voice, enterprise video and data) on demand. This architectural view is still defined at a high-level view and does not detail the mapping of the functions sets to different levels.

Kayvartya et al. [4] have proposed a comprehensive IoV five-layer architecture, to support an enriched set of vehicular communications, in addition to traditional V2V, V2R/V2I, i.e., *Vehicle-to-Personal* devices (V2P) and *Vehicle-to-Sensors* (V2S). Each particular IoV communication type can be enabled using a different WAT, e.g., IEEE WAVE for V2V and V2R, Wi-Fi and 4G/LTE for V2I, CarPlay/NFC (*Near Field Communications*) for V2P and WiFi for V2S. The system includes vehicles and *Road Side Units* (RSU), but also other communication devices. Embedding such a large range of devices makes the IoV more complex, (compared to VANET), but more powerful and market oriented. Three architectural planes are defined: *management*, *operation* and *security*. This allows mapping of various existing protocols and functions (e.g., taken from ITS) to architectural layers. The network model is composed of three functional entities: *client, connection* and *cloud.*

The five layers in [4] are: *perception, coordination, artificial intelligence, application* and *business.* The perception layer (PL) includes the traditional physical layer functions and some additional for sensing and actuating actions. The coordination layer (CL) represents a virtual universal network coordination entity for heterogeneous network technologies (WAVE, Wi-Fi, 4G/LTE, satellites, etc.). The artificial intelligence layer (AIL) is represented by a generic virtual cloud infrastructure, working as an information processing and management centre. It stores, processes and analyzes the information received from the lower layer and then takes decisions. Its major components are: *Vehicular Cloud Computing (VCC), Big Data Analysis (BDA)* and *Expert System.* The AIL should meet the requirement of applications and services of the AL. The application layer (AL) contains smart applications (e.g., for traffic safety and efficiency, multimedia-based infotainment and web-based utility). The *business* layer (BL) includes IoV operational management functions, basically related to business aspects.

The above 5-layer architecture does not discuss how to distribute computation intelligence between a central cloud and fog/edge units in combined cloud-fog/MEC solution. Neither SDN-like control nor NFV implementation possibilities are discussed.

F. Yang et al. [5] suggest an IoV architecture, by considering the driver-vehicle-environment coordination. IoV is defined as an open converged network system (controllable, manageable, operational, and trustable) based on multi-human, multi-machine, multi-vehicle, and environment coordination. It senses, recognizes, transmits, and computes the large-scale complex static/dynamic information of human, vehicle, network communication and road traffic infrastructure, using advanced ICT technology.

Four layers are defined: the *environment sensing and control* layer, *network access* and *transport* layer, *coordinative computing control* layer, and *application* layer. The work also summarizes the core technologies of each layer. The coordinative computing control layer has a special role to coordinate among human-vehicle-environment. The application layer provides various types of services and is open (i.e., it can support novel services and business operating modes). The types of services can be: closed (related to the specific industry applications) or open (i.e., various existing open applications, such as real-time traffic services provided by Internet service providers or by third party providers). Neither MEC/FC nor SDN/NFV approaches are discussed in [5]. The homogeneity of sub-layers is rather low in terms of their components. No architectural split in planes is proposed; so, it is rather difficult to see how to map different already developed functions and protocols (coming from ITS, WAVE, etc.) to the layers of this architecture; this seems to be still an open issue.

A seven-layer (6+1) IoV architecture is proposed by Contreras-Castillo et al. in [6], to support collaboration between multi-users, multi-vehicles, multi-devices (sensors, actuators, mobile devices, access points), multi-communication models (point to point, multi-point, broadcast, geo-cast) and multi-networks (wireless or wire networks with various technologies like Wi-Fi, Bluetooth, WiMAX, 3G, 4G/LTE, etc.). The layers are (bottom-up list): User interaction (lowest layer), Data acquisition, Data filtering and pre-processing, Communication, Control and management, Business (highest layer). A macro-layer is defined and named Security; it is rather a cross layer entity. The cloud services are located at business level (as vehicular cloud computing) while we believe that a more natural placement is below to application layer. Some mixture of "layers" and "plane" notions is apparent; there is a lack of

 ISBN: 978-1-61208-662-0

orthogonality of different "layers". The architecture does not emphasize any MEC/Fog solutions or integration of SDN/NFV approach.

Here, it is considered that Kayvartya et al. [4] architecture is a good IoV model, enough flexible to accommodate computing technologies like MEC and FC.

IV. MEC AND FOG SOLUTIONS INTEGRATED IN IOV

This section presents IoV relevant systems which include MEC and Fog approaches to identify some pros and cons of each solution in the IoV context.

A MEC-based model of a vehicular network is developed by K. Zhang, et al., in [20]. The architectural levels are: *Virtual Computation Resource Pool*- incorporating the network and cloud resources outside the MEC; *MEC level* – implemented as MEC servers placed in the RAN; *RSUs units* placed on the roads; mobile units (vehicles). A special focus is on the computation off-loading process, to preserve the service continuity in a mobile environment. Vehicles in transit may pass through several RSUs and MEC servers during the task-off-loading process, and they can off-load their computation task to any MEC servers that they can access. Two methods are possible: selection of the target MEC servers or selecting (for a while) of a new path from the mobile vehicle to the same MEC server (keeping as much as possible the same serving MEC server in order to avoid too frequent moving of virtual machines).

J. Liu et al., [21] propose an SDN-enabled network architecture assisted by MEC, while integrating different types of access technologies. The architectural components are (top-down hierarchical list): Remote Data Center; Backbone network, Regions (each one contains MEC servers collocated with an SDN controller, BS and mobiles organized in VANETs). The MEC servers can inter-communicate via a mesh of fixed network link. This architecture has an SDN-like control, comprising three planes (Data, Control and Application) each including typical functions. The Data Plane (DPl) includes SDN-"switches" (VANET, BS, Ethernet); lower layer technologies (IEEE 802.11p, LTE/5G, Wire NIC, etc.). The Control Plane (CPl) has two sub-layers: lower sub-layer with functions such as Position/Channel sensing, Flow table management, Forwarding strategy; upper sublayer: Trajectory prediction, Interface sensing, Radio Resource control, Traffic redirection. The Application Plane (APl) (in the SDN semantics) includes Topology management, Resource Management, Traffic Offload, SDN controller.

The interface between CPl and DPl is based on extended OpenFlow or other similar protocol. The details of the layer mapping on SDN/NFV and fog/edge approach are not discussed. The MEC server is considered as belonging to the infrastructure and it is transparent to the client (vehicle). The client can request services from or deliver packets to a remote cloud server. If the requested service is deployed on an MEC server, then the BS redirects it to the MEC server. MEC usually stores recent traffic data and responds to real-time events. RSUs collect the real-time road conditions and deliver them to the MEC server (via BS). The traffic data should be pre-processed by the MEC server before they are delivered to the remote cloud server by means of data synchronization. The remote cloud server stores traffic data permanently and makes a traffic prediction based on real-time and historical data.

K. Zheng et al. [22] propose an IoV architecture called software-defined heterogeneous vehicular network (SERVICE), based on Cloud-RAN technology suitable for the dynamic nature of heterogeneous VANET functions and various applications. A multi-layer Cloud-RAN multi-domain is introduced, where resources can be exploited as needed for vehicle users. The system is hierarchically organized (three levels of clouds are defined: remote, local and micro clouds) and virtualization techniques (offering flexibility) are considered for implementation. However, this work does not map the architecture on specific MEC or fog solutions. The high-level design of the soft-defined HetVNET is presented. The SDN control is organized on two levels (one primary controller and several secondary controllers; each one of the latter controls a given service area). A complete layered functional IoV architecture is not in the paper scope.

A Fog-SDN architecture called FSDN is proposed for advanced VANET by Truong et al. [23], for V2V, V2I and Vehicle-to-Base Station communications. The Fog computing brings more capabilities for delay-sensitive and location-aware services. The SDN components (hierarchically top-down listed) are: SDN Controller (it controls the overall network behavior via OpenFlow – interfaces; it also performs Orchestration and Resource Management activities for the Fog nodes); SDN RSU Controller (RSUC) (controlled by the central SDN controller; each RSUC controls a cluster of RSUs connected to it through broadband connections. The RSUC can forward data, and store local road system information or perform emergency services. From Fog perspective RSUC are fog devices); SDN RSU (it is also a Fog device); SDN Wireless Nodes (vehicles acting as end-users and forwarding elements, equipped with OBU); The system also contains Cellular Base Station (BS) performing traditional functions (they are SDN-controlled via OpenFlow protocol and can also offer Fog services). This study does not map the functions on a full layered architecture.

Kai et al. [23] work presents an overview of Fog–SDN solution for VANET and discuss several scenarios and issues. It is shown that a mixed architecture Fog-SDN can be powerful and flexible enough, to serve future needs of IoV.

OpenFog Consortium presents in [15] a complex system, cloud-fog-based, for Transportation Scenario (Smart Cars and Traffic Control). They took into account the high amount of data generated (multiple terabytes of data every day from the combinations of light detection and ranging (LIDAR), global positioning systems (GPS), cameras, etc.). So, a combined cloud-fog computing approach is required; the system can be supported by OpenFog Reference Architecture. Figure 3 shows an overview of an intelligent highway application of the OpenFog RA.

 ISBN: 978-1-61208-662-0

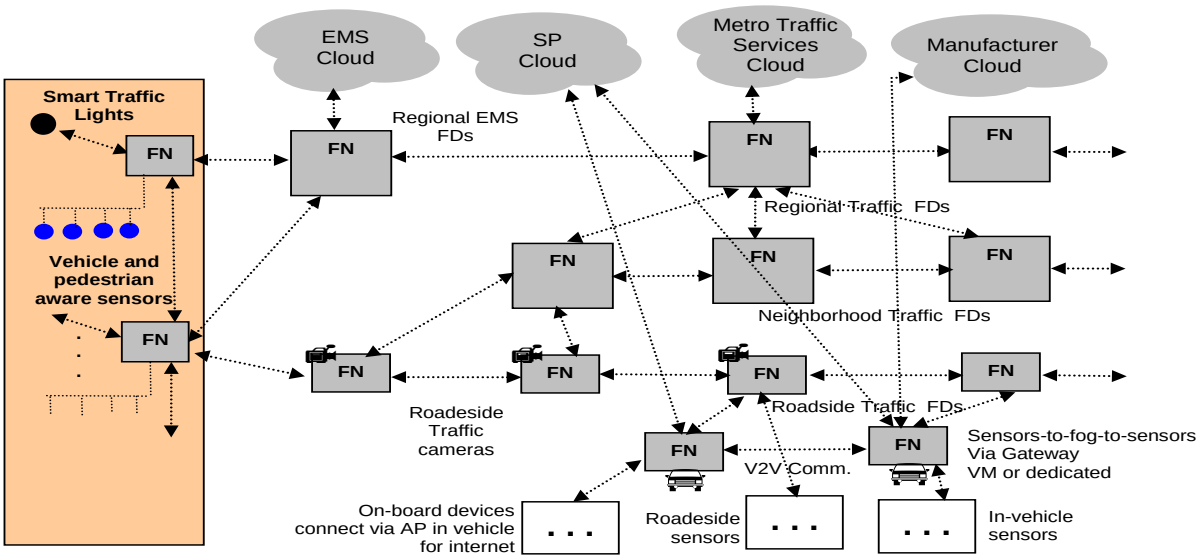

Figure 3. General cloud-fog-based example system for transportation scenario (smart cars and traffic control) [adapted from [14]]
EMS- Element Management System; SP- Service Provider; FD- Fog Device; FN- Fog Node

The system is based on a fog environment containing a rich set of interactions among multiple fog and multiple cloud domains. The fog architecture is hierarchical and distributed which is an important advantage. Some important capabilities of fog technology while applied in IoV domain are illustrated: a rich set of interactions among multiple fog and cloud domains, including Element Management Systems (EMS), service provider (SP), metro traffic services, and system manufacturer clouds; mobile fog nodes supporting V2V, V2I and V2X interactions; multiple fog networks owned and operated by different authorities providing similar (and different) functionality; multi-tenancy across fog nodes allows to consolidate multiple fog networks; both private and public fog and cloud networks used by a single end point device. The system includes several types of sensors (and actuators) referred to as "things." Things include roadside and on-vehicle entities, to provide data, so that the various systems (lights, cars, etc.) can carry out their functions (e.g., vehicle driving autonomously). Smart transportation systems also manage the actuators that control parts of the infrastructure, such as traffic signals, gates, and digital signs. The vehicles connect to the cloud and a hierarchy of fog nodes that service the autonomous vehicle or traffic control systems.

The vehicle itself can be a mobile FN, communicating with other FNs as they become available. The mobile fog node can perform all required in-vehicle operations autonomously (if it cannot connect to other FNs or to the cloud). In-vehicle fog nodes provide services including infotainment, Advanced Driver Assistance Systems (ADAS), autonomous driving, collision avoidance, navigation, etc. The Transportation Fog Network has a three-level hierarchy of FNs; the first is the infrastructure (roadside) fog nodes. The roadside fog sensors collect data from other devices such as roadside cameras. The FNs perform some local analysis for local actions. Data from the first level is aggregated and sent up to the second and third levels of the hierarchy—neighborhood and regional fog nodes—for further analysis and distribution.

The above use case demonstrates the goal of the OpenFog RA for smart cars and traffic control, i.e., to ensure an open, secure, distributed, and scalable architecture that optimizes real time capabilities within a multi-supplier ecosystem. The transportation example shows a complex system of autonomous things and infrastructure generating massive amounts of data. This use case highlights the need for fog computing to enable safe and effective operations in IoT, 5G, AI and other advanced scenarios.

V. MEC VERSUS FOG IN IOV

The previous section selected some relevant examples of architectures and systems to illustrate the MEC/FC usage in the context of IoV. Note that also other proposals are published in the literature, both for MEC and Fog. While not all aspects of the architectural could be discussed in few examples, some distinctive features can be emphasized.

To a question like "selection of Mobile Edge computing, versus Fog computing for IoV system" one cannot have a unique general answer, given the facts that - both architectures and technologies are edge-oriented; they have as a main idea to move the cloud computing-like capabilities to the networks edges in order to obtain advantages mentioned in section II. The second reason is that a realistic selection could depend significantly on the IoV services needed - out of a large set described in Introduction section.

MEC/FC have quite a lot of common characteristics like: low latency; support for real time interactions, location awareness and mobility and large number of server nodes; geographical distribution proximity to the end devices (single network hop or few hops); service location at the edge of the local network; various working environment outdoor (streets, base stations, etc.) or indoor (houses, cafes,

 ISBN: 978-1-61208-662-0

etc.); wireless communication access: WLAN, WiFi, 3G, 4G, ZigBee, etc., or wired communication (part of the IP networks); weak dependence on the quality of core network; low bandwidth costs and energy consumption. However, both have weak computation and storage capabilities, which raises a need for them to cooperate with CC.

Both MEC and FC can benefit from technologies like SDN and NFV in different architectures. Both MEC and Fog can be compliant with the layered architectures described in Section III.

TABLE I. MEC VERSUS FOG DIFFERENCES

Criterion	MEC	Fog computing
Placement of node devices	Servers running in Base Stations Network Controller/Macro Base Station	Anywhere - between end devices and cloud: Routers, Switches, Access Points, Gateways
Compute Distribution and Load Balancing	Employ a strategy of placing servers, apps or small clouds at the edge	Broader architecture and tools for distributing, orchestrating, and securing resources and services across networks
Software Architecture	Mobile Orchestrator based (strongly specified)	Fog abstraction layer based (only partially specified)
Standardization/ specifications	ETSI/	/OpenFog Consortium
Context awareness	High	Medium
Proximity	One hop	One or multiple hops
Access Mechanisms	Mobile networks: 3G/4G/5G	Wi-Fi, Mobile networks, etc.
Virtualization and management mechanisms	Strongly specified by ETSI (NFV framework)	Larger view of virtualization. In progress at OpenFog Consortium
Hierarchical structure of the overall system	Possible	Yes: multiple levels of cooperating nodes, supporting distributed applications
Horizontal scalability	Medium	High
Internode Communication	Possible - between Mobile Edge Hosts	Native support for communication between Fog nodes
Communication with Cloud	Possible	Fog-cloud is usually considered necessary
Modular architecture with multiple access modes	Edge deployments are typically based on gateways with fixed functionality. However they can be made more flexible and dynamic by using NFV.	Highly modular architecture; every Fog node has exactly the resources its applications need; it can be dynamically configured.
Topology of server nodes	Less flexible (limited by RAN spread)	More general and very flexible
5G compliant specifications	Full compliancy	Work in progress

There are also differences between FC and MEC from several points of view (see also [25][26]), as summarised in Table I. The presented criteria can serve in order to make a selection of MEC/FC in a specific IoV use case.

The MEC/FC paradigm can offer, in the context of IoV, support for a large variety of applications, use-case scenarios, and heterogeneous end devices. On the other side, different use cases and applications might have their own set of requirements and trade-offs which can determine which solution between MEC or FC is the appropriate choice.

Note that, for a given set of use cases to be provided by an IoV system, the problem of selecting MEC or FC approach is multi-criteria one. Among the parameters for selection there are those presented in Table I, where appropriate weights should be assigned to them.

Last but not least, one has to consider the strong effort for cooperation between different organizations, towards a convergence of vision in the domain of edge computing (including MEC, Fog, Cloudlets, etc.)

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a preliminary comparative view of Mobile Edge Computing and Fog computing, used as support technologies in Internet of Vehicles, from architectural and technological point of view. A comparison of the technologies has been performed in Section V, identifying the common MEC/FC characteristics and also differences which could be considered when selection of MEC/FC has to be done (depending on the target use cases) in order to implement a given IoV system.

The conclusion of this study is that given the large variety of target IoV systems and use case envisaged, there is no winner MEC versus FC technology, but a selection should be done for each specific case in a *multi-criteria mode*. Different priorities can be assigned to criteria, depending on specific needs of use and business case.

However, some general guidelines can be expressed. MEC approach is more restricted than FC in terms of network dimension and vertical hierarchy, but the IoV development based on MEC can benefit form: detailed elaborated specifications coming from ETSI for MEC; powerful virtualization support defined by NFV technology which is fully compliant with MEC; SDN/NFV approach can be naturally applied in MEC implementation; resource management, mobility and task offloading are aspects better defined in terms of solutions in MEC framework than in fog computing.

Fog computing solutions for IoV have the advantage of being more general in terms of hierarchization, flexibility, geographical span, extension on the core network of FC capabilities. However, if selecting a fog computing solution for IoV then additional challenges should be considered [14], in comparison with traditional fog computing: the edge nodes can be highly mobile causing possible intermittent loss of connection to the remote cloud servers; the computation can be based on vehicular control engines, and therefore accuracy and safety criticality must be ensured; access control is important for vehicular fog computing environments and should be enforced sometimes in real-

 ISBN: 978-1-61208-662-0

time mode to prevent delays of some critical decision related to traffic; in a vehicular environment, failure or sporadic behaviors of a few sensor nodes may affect the control decisions taken over a fog (ensuring correctness of the local computation needs to be ensured for intelligent or autonomous vehicles).

Future work should be done to detail some more specific and also quantitative problems for both MEC and Fog approaches in IoV, like resource management and scheduling in the virtualization context, the computation tasks off-loading (in a mobile context) problems to assure the service continuity, MEC/FC in network slicing context, security and privacy aspects, multi-tenancy capabilities, etc.

REFERENCES

[1] ETSI EN 302 665 V1.1.0 European Standard Telecommunications series, "Intelligent Transport Systems (ITS);Communications Architecture" (2010-07).

[2] Y. Li, "An Overview of the DSRC/WAVE Technology", https://www.researchgate.net/publication/288643779, 2012, [Retrieved: May, 2017].

[3] S. Sultan, M. Moath Al-Doori, A.H. Al-Bayatti, and H.Zedan "A comprehensive survey on vehicular Ad Hoc Network", J.of Network and Computer Applications, Jan. 2014, https://www.researchgate.net/publication/259520963, [Retrieved: August, 2018].

[4] O. Kaiwartya, A.H. Abdullah, Y. Cao, A. Altameem, and M. Prasad, "Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects", IEEE Access, Special Section on Future Networks, Architectures, Protocols and Applications, Vol. 4, pp.5536-5372, September 2016.

[5] F. Yang, J. Li, T. Lei, and S. Wang, "Architecture and key technologies for Internet of Vehicles: a survey", Journal of Communications and Information Networks, Vol.2, No.2, June 2017, DOI: 10.1007/s41650-017-0018-6.

[6] J. C. Contreras-Castillo et al., "A seven-layered model architecture for Internet of Vehicles", Journal of Information and Telecommunications , Vol. 1, No. 1, pp. 4–22, 2017.

[7] Y. Fangchun, W.Shangguang, L. Jinglin, L. Zhihan, and S.Qibo, "An overview of Internet of Vehicles", China Commun., vol. 11, no. 10, pp. 115, October 2014.

[8] F. Bonomi, "The smart and connected vehicle and the Internet of Things", San José, CA: WSTS, 2013, https://tf.nist.gov/seminars/WSTS/PDFs/1-0_Cisco_FBonomi_ConnectedVehicles.pdf, [August, 2018].

[9] K. Tocze and S. Nadjm-Tehrani, "A Taxonomy for Management and Optimization of Multiple Resources in Edge Computing", arXiv: 1801.05610v1 [cs.DC] 17 Jan 2018, [Retrieved: May, 2018].

[10] N. Mohan and J. Kangasharju, "Edge-Fog Cloud: A Distributed Cloud for Internet of Things Computations", arXiv:1702.06335v2 [cs.DC] 7 Mar 2017, [Retrieved: March, 2018].

[11] "Mobile edge computing (MEC); Framework and reference architecture", ETSI, Sophia Antipolis, France, Mar. 2016. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf [Retrieved: January, 2018].

[12] T. Taleb, K.Samdanis, B.Mada, H. Flinck, S.Dutta, and D.Sabella, "On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration", IEEE Comm. Surveys &Tutorials, Vol19, No.3, pp.1657-1681, 2017.

[13] F. Bonomi, R. Milito, J. Zhu, and Sateesh Addepalli, "Fog Computing and Its Role in the Internet of Things", August 2012, http://conferences.sigcomm.org/sigcomm/2012/paper/mcc/p13.pdf, [Retrieved: August, 2018].

[14] S.B.Nath, H.Gupta, S.Chakraborty and S.K Ghosh, "A Survey of Fog Computing and Communication: Current Researches and Future Directions", https://arxiv.org/pdf/1804.04365.pdf, [Retrieved: June, 2018].

[15] OpenFog Consortium (2015), OpenFog Reference Architecture for Fog Computing,

[16] http://www.openfogconsortium.org/resources/#definition-of-fog-computing, [Retrieved: August, 2018].

[17] B. N. Astuto, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks", Communications Surveys and Tutorials, IEEE Communications Society, (IEEE), 16 (3), pp. 1617 – 1634, 2014.

[18] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network Function Virtualisation: Challenges and Opportunities for Innovations". IEEE Communications Magazine, pp. 90-97, February 2015.

[19] E. Bacarelli, P. G. Vinueza Naranjo, M. Scarpinti, M. Shojafar, and J. H. Abawaji, "Fog of Everything: Energy-Efficient Networked Computing Architectures, Research Challenges, and a Case Study", IEEE Access, Vol.5, 2017, pp. 9882-9910.

[20] T. Taleb and A. Ksentini, "Follow me cloud: Interworking federated clouds and distributed mobile networks" *IEEE Network*, vol. 27, no. 5, pp. 12_19, Sep./Oct. 2013.

[21] K. Zhang, Y. Mao, S. Leng,Y. He, and Y. Zhang, "Mobile-Edge Computing for Vehicular Networks- A Promising Network Paradigm with Predictive Off-Loading", IEEE vehicular technology magazine, pp. 36-44, June 2017.

[22] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing", IEEE Communications Magazine, pp. 94-100, July 2017.

[23] K. Zheng, L. Hou, H. Meng, Q. Zheng, and N. Lu, "Soft-Defined Heterogeneous Vehicular Network: Architecture and Challenges", IEEE Network, vol. 30, pp. 72-79, July/August 2016.

[24] K. Kai, W. Cong, and L. Tao, "Fog computing for vehicular Ad-hoc networks: paradigms, scenarios, and issues", The Journal of China Universities of Posts and Telecommunications, www.sciencedirect.com/science/journal/10058885, http://jcupt.bupt.edu.cn, 23(2), pp. 56–65, April 2016, [Retrieved: August, 2018].

[25] N. N. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular ad hoc network with fog Computing", Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM'15), May 2015, Ottawa, Canada. Piscataway, NJ, USA: IEEE, pp. 1202–1207, 2015.

[26] K. Dolui and S. K. Datta, "Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing", Global Internet of Things Summit (GIoTS), 2017, https://ieeexplore.ieee.org/document/8016213/, [Retrieved: January, 2018].

[27] OpenFog Consortium, "10 ways fog computing extends the edge", August 21, 2017, https://www.rtinsights.com/10-ways-fog-computing-extends-the-edge/, [Retrieved: August, 2018].

 ISBN: 978-1-61208-662-0